ROBOFUN.RO

LECȚIA VI

CURS GRATUIT

ARDUINO ȘI ROBOTICĂ

Senzori Forţa

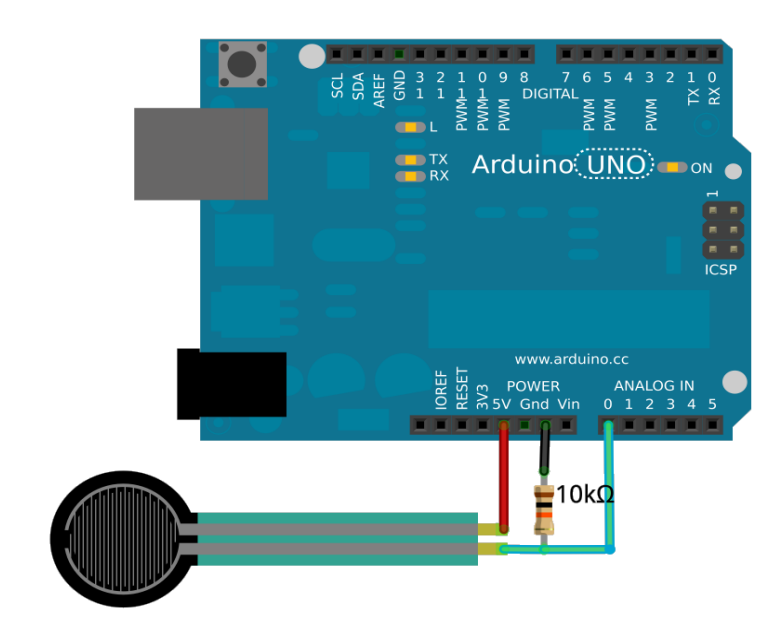
Proiect :

Toba cu senzori piezo

Senzori de Apasare

Senzorii de apasare ofera o modalitate simpla de a sesiza forta care actioneaza asupra lui. Sunt disponibili in mai multe dimensiuni. Din punct de vedere al functionarii, un astfel de senzor poate fi vazut ca un potentiometru rezistiv care isi schimba valoarea proportional cu forta care este aplicata asupra lui. Pentru a-l utiliza impreuna cu Arduino, cea mai simpla abordare este sa-l conectam in serie cu un rezistor de 10K, si sa folosim principiul divizorului de tensiune pentru a citi caderea de tensiune pe rezistorul de 10K. In acest mod, atunci cand senzorul de apasare isi modifica rezistenta, se va modifica si curentul prin circuit, si implicit si caderea de tensiune pe rezistorul de 10 K (pe care o citim noi pe un port analogic).

Codul sursa este extrem de simplu, nu face altceva decat sa citeasca valoarea de pe pinul analogic A0 si sa o afiseze in Serial Monitor. Valorile afisate in Serial Monitor in aceasta situatie sunt valori de ordin calitativ ("acum apas mai tare decat am apasat data trecuta"), fara a fi etalonate neaparat in newtoni. Senzorul este capabil de o precizie de aproximativ 10 %, lucru pe care trebuie sa-l iei in considerare daca vrei sa construiesti un cantar de farmacie.



Arduino 5V	PIN1 senzor
Arduino GND	PIN1 Rezistor 10K
PIN2 Rezistor 10K	Pin2 Senzor
Arduino Analog0	PIN2 Rezistor 10K

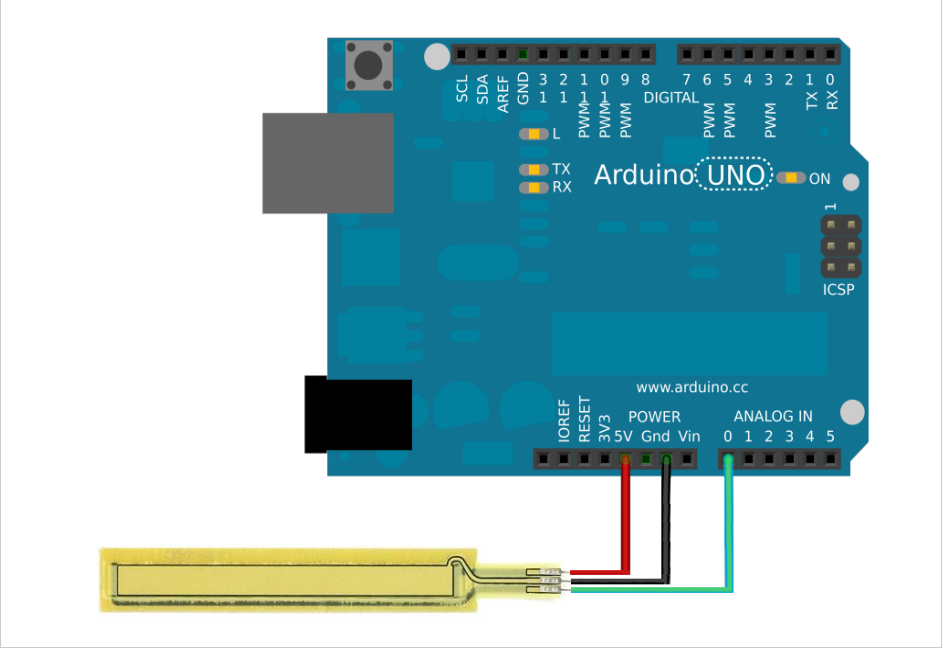
```
void setup() {
 Serial.begin(9600);
}

void loop() {
  int nivelForta = analogRead(0);
  Serial.print("Nivel Forta: ");
  Serial.println(nivelForta,DEC);
}
```

Cand lipesti firele pe terminalele senzorului, fii foarte atent. Pastreaza cat mai putin timp letconul in contact cu terminalele senzorului, altfel risti sa topesti plasticul din jur (ceea ce va duce la distrugerea senzorului). Cea mai buna metoda este sa topesti mai intai un pic de fludor pe terminal, apoi separat topesti un pic de fludor pe fir, apoi le pui in contact si finalizezi lipitura. Daca din diverse motive fludorul nu adera rapid pe terminalul senzorului (1-2, maxim 3 secunde), atunci ridica letconul si incearca din nou peste 30 de secunde cand s-a racit terminalul.

Senzor de Atingere HotPot

Senzorii de atingere determina punctul in care sunt apasati cu degetul. Practic, senzorul se comporta ca un rezistor variabil care isi modifica rezistenta in functie de punct in care este apasat. Limitele sunt intre 100 ohm si 10 KOhm.

Utilizarea cu Arduino este la fel de simpla ca mai sus, doar ca nu mai ai nevoie de rezistorul extra. Schema de conectare este mai jos.



Arduino 5V	PIN1 Senzor
Arduino GND	PIN3 Senzor
Arduino Analog0	PIN2 Senzor

```
void setup() {
 Serial.begin(9600);
}

void loop() {
  int nivelForta = analogRead(0);
  Serial.print("Nivel Forta: ");
  Serial.println(nivelForta,DEC);
}
```

Senzor de Indoire

Senzorii de indoire ofera o modalitate simpla de a sesiza gradul de indoire al unui element din mediu. Sunt disponibili in mai multe variate de dimensiuni.

Din punct de vedere al functionarii, un astfel de senzor poate fi vazut ca un potentiometru rezistiv care isi schimba valoarea corespunzator cu gradul de indoire la care este supus. Pentru a-l utiliza impreuna cu Arduino, cea mai simpla abordare este sa-l conectam in serie cu un rezistor de 10K, si sa folosim principiul divizorului de tensiune pentru a citi caderea de tensiune pe rezistorul de 10K. In acest mod, atunci cand senzorul de apasare isi modifica rezistenta, se va modifica si curentul prin circuit si implicit si caderea de tensiune pe rezistorul de 10 K (pe care o citim noi pe un port analogic).

Codul sursa este extrem de simplu, nu face altceva decat sa citeasca valoarea de pe pinul analogic A0 si sa o afiseze in Serial Monitor. Valorile afisate in Serial Monitor sunt proportionale cu gradul de indoire al senzorului. Senzorul sesizeaza si directia de indoire (in sensul ca daca il indoi spre stanga, valoarea rezistentei electrice creste, iar daca il indoi spre dreapta, valoarea scade). Astfel, poti determina exact atat directia indoirii, cat si cat de mult este indoit senzorul.

Pentru etalonare, cel mai simplu este sa faci incercari repetate. Astfel, prima data vei lasa senzorul liber si vei urmari ce valoare ai in Serial Monitor. Acea valoare este valoarea pe care o vei folosi mai departe in codul tau ca sa detectezi cand senzorul este perfect drept.

Arduino 5V	PIN1 senzor
Arduino GND	PIN1 Rezistor 10K
PIN2 Rezistor 10K	Pin2 Senzor
Arduino Analog0	PIN2 Rezistor 10K

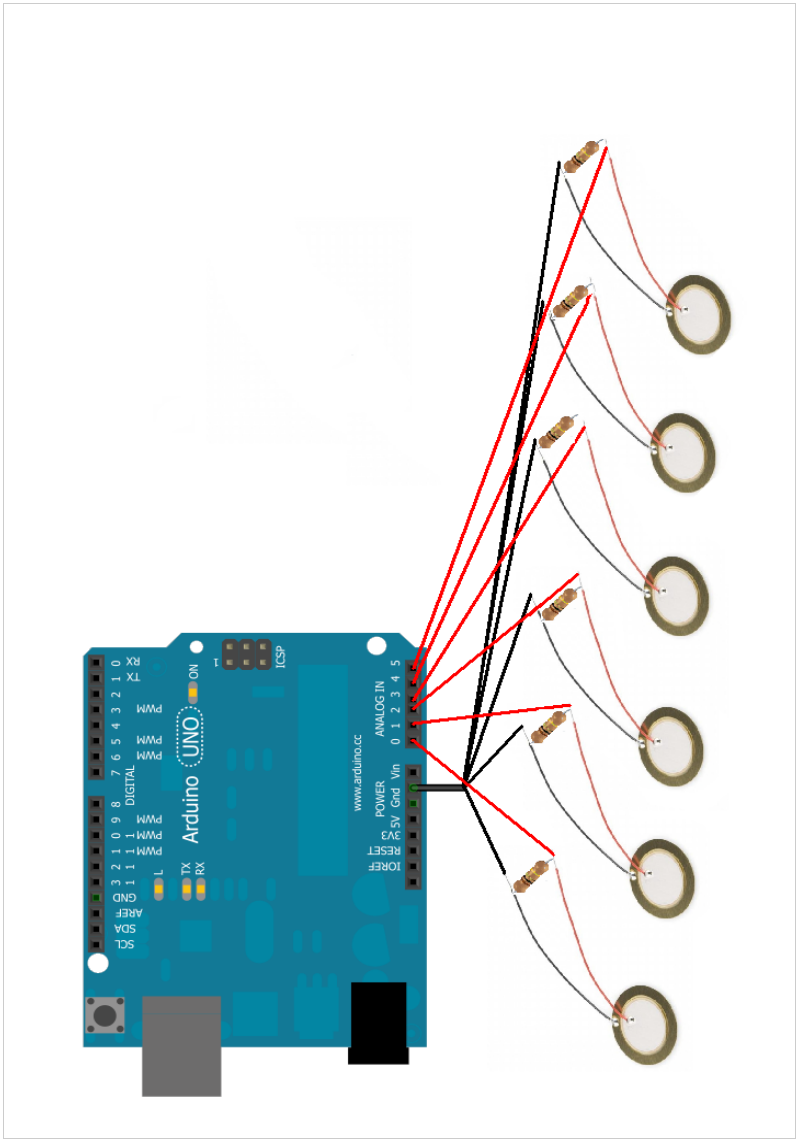
```
void setup() {
 Serial.begin(9600);
}

void loop() {
  int nivelForta = analogRead(0);
  Serial.print("Nivel Indoire: ");
  Serial.println(nivelForta,DEC);
}
```

Senzor Piezo

Senzorul piezo ofera o modalitate foarte simpla de a sesiza vibratiile din mediul ambiat. Functionarea lui este extrem de simpla, ori de cate ori senzorul este supus unei vibratii mecanice, genereaza o diferenta de potential intre cele doua borne ale sale. Mai departe, daca vom conecta firul negru (GND) la pinul GND Arduino si firul rosu (semnal) la unul dintre pinii analogici Arduino (sa spunem ca alegem pinul A0), vom sesiza o citire pozitiva pe pinul A0 ori de cate ori senzorul vibreaza.

Daca ne-am opri doar aici (si poti incerca sa faci doar montajul descris mai sus si sa vezi cum se comporta), vei sesiza ca odata ce senzorul a vibrat la un moment dat, valoarea citita pe portul analogic va ramane la valoarea citita in momentul vibratiei (sau va scadea foarte incet spre 0). Ca sa fortam o revenire la 0 dupa ce vibratia mecanica a disparut, se conecteaza un rezistor cu valoare mare (de exemplu, 1 Mega Ohm) intre cele doua fire ale senzorului. Astfel, diferenta de potential produsa de senzor este descarcata prin rezistor intr-un anumit interval de timp (care depinde de valoarea rezistorului – cu cat rezistorul este mai mare, cu atat valoarea citita ajunge la 0 mai rapid si cu cat rezistorul este mai mare, cu atat valoarea citita scade mai incet). Din teste, 1 Mega Ohm pare a fi o valoare suficient de buna.

Codul sursa este extrem de simplu, nu face altceva decat sa citeasca valoarea de pe pinul analogic A0 si sa o afiseze in Serial Monitor. O schema cu conectarea pentru sase senzori piezo vei gasi mai jos. Firul negru al fiecarui senzor se conecteaza la pinul GND Arduino, firul rosu al fiecarui senzor se conecteaza la cate un pin analogic, iar intre firul rosu si firul negru se conecteaza un rezistor de 1 Mega Ohm.

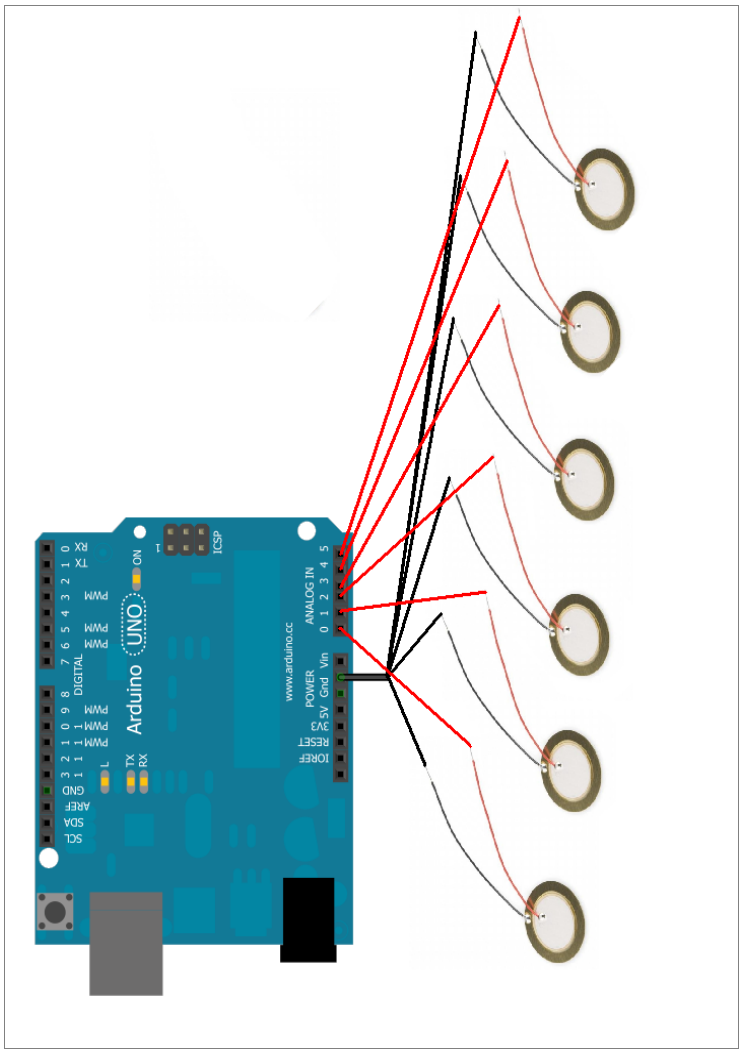


```
void setup() {
 Serial.begin(9600);
}

void loop() {
  int nivelVibratie = analogRead(0);
  Serial.print("Nivel Vibratie: ");
  Serial.println(nivelVibratie,DEC);
}
```

Toba cu senzori piezo

Un senzor piezo este capabil sa detecteze vibratiile. Ori de cate ori este facut sa vibreze, la bornele lui apare o tensiune electrica. Aceasta tensiune electrica este detectata folosind o placa Arduino, care placa Arduino comanda catre Music Instrument Shield generarea unui sunet corespunzator unui instrument de percutie, la alegerea ta. Obtii astfel o toba virtuala controlata de Arduino.

Schema de conectare este simpla. Senzorul piezo are doua fire, unul negru (masa) si unul rosu. Intre cele doua fire se inregistreaza diferenta de tensiune generata in cazul vibratiilor. Vei conecta toate firele negre impreuna la pinul GND al Arduino si fiecare fir rosu la cate un pin analogic. Daca faci doar atat, tensiunea generata de senzorul piezo va scadea extrem de incet (dupa ce senzorul a vibrat la un moment dat, vei avea aceeasi valoare pe portul analogic Arduino, chiar daca senzorul nu mai vibreaza). Poti incerca acest lucru cu codul de mai jos.

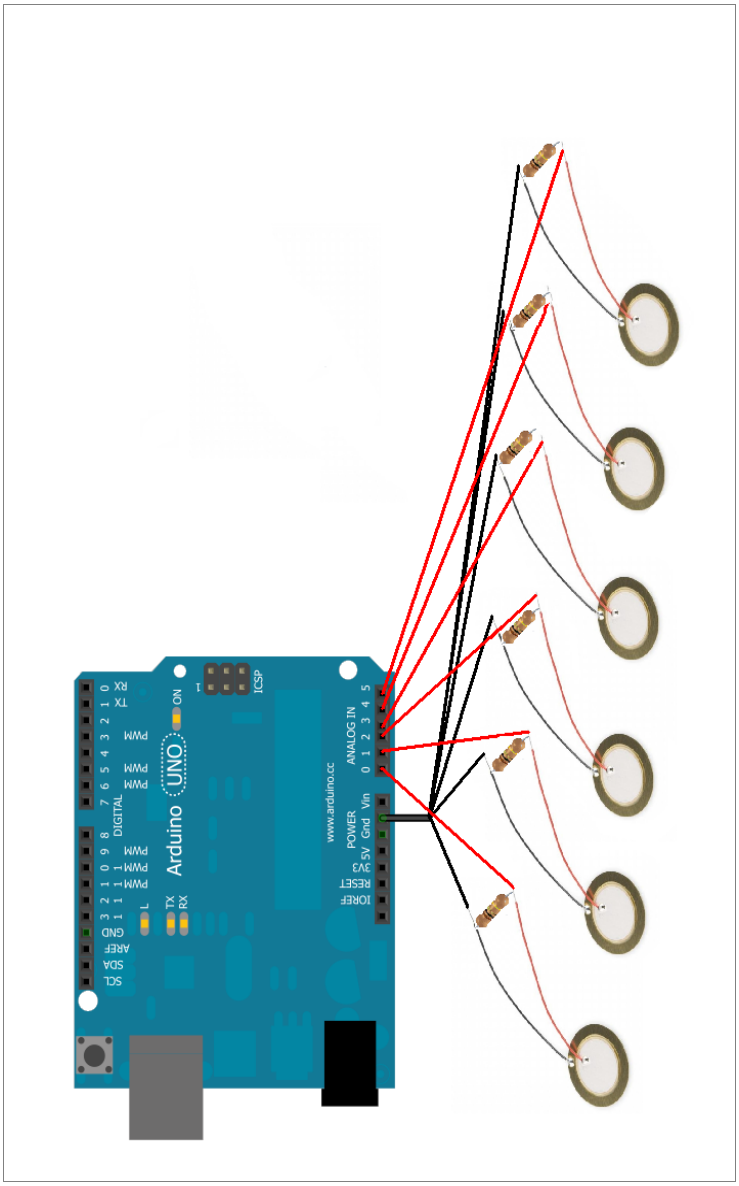



```
void setup() {
      Serial.begin(9600);
}
void loop() {
      int v = analogRead(0);
      Serial.println(v);
}
```

Daca vei misca (sau vei atinge cu degetul) senzorul piezo conectat la portul analogic zero, vei vedea ca valoarea afisata in interfata de debug se modifica, si apoi scade foarte foarte incet inapoi la zero.

Ca sa fortam valoarea sa scada spre zero mai rapid, vom cupla in paralel cu fiecare senzor un rezistor de valoare mare (de exemplu, un megaOhm). Unul dintre pinii rezistorului se va conecta la firul rosu, iar celalalt pin la firul negru al senzorului piezo. In acest fel, tensiunea generata de senzor se va descarca incet prin rezistor, si dupa o secunda, vom avea iarasi o citire zero pe senzor. Cu cat alegi un

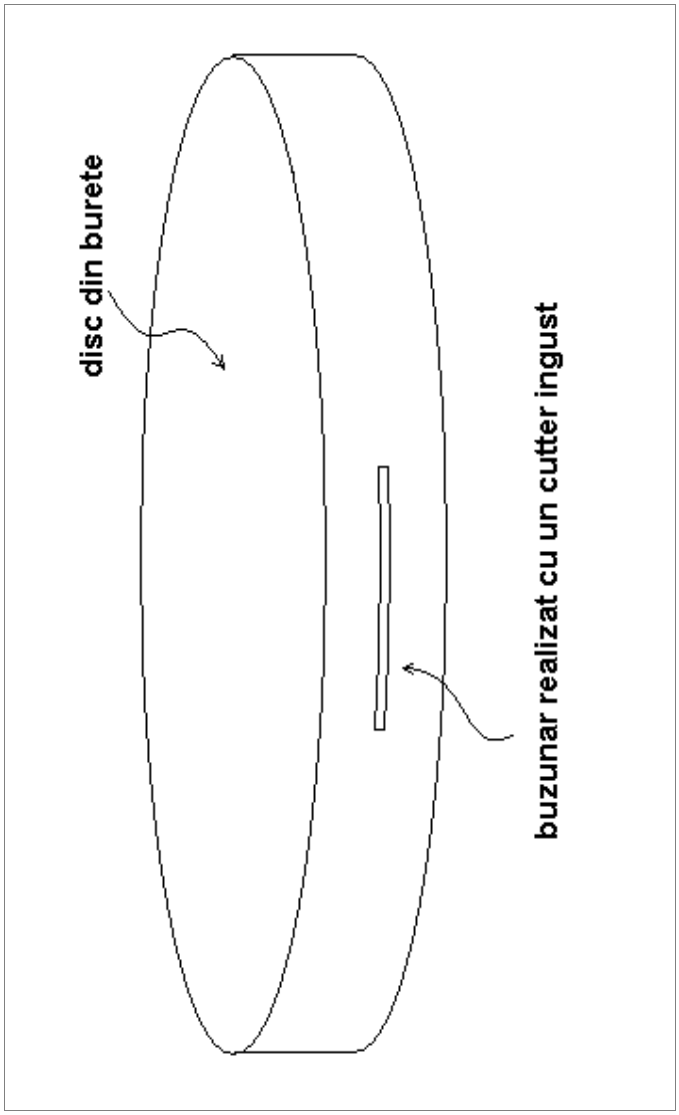
rezistor mai mare, cu atat valoarea citita de Arduino va persista mai mult. Cu cat rezistorul este mai mic, cu atat raspunsul va fi mai rapid. Poti verifica acest lucru cu programul de mai jos.




```
#define PRAG 3
void setup() {
      Serial.begin(9600);
}
void loop() {
      int v = analogRead(0);
      if (v > PRAG) {
            Serial.println(v);
      }
}
```

Rolul instructiunii *IF* este de a-ti permite sa vezi valorile modificate. Daca eliminam if-ul, vei vedea in debug foarte multe valori zero si doar pentru o perioada foarta scurta vor aparea valori mai mari ca zero. Datorita faptului ca perioada este extrem de scurta, cel mai probabil nu vei reusi sa vezi nimic. Cu instructiunea if, vei vedea practic doar valorile mai mari ca zero in interfata de debug.

Pentru montarea efectiva, iti recomand sa cumperi din Hornbach, BricoStore, Praktiker si alte magazine de profil o bucata de burete gros (in jur de 7 - 10 cm grosime, se foloseste in mod normal pentru saltele). Din acest burete taie discuri cu diametru mai mic si altele cu diametru mai mare. Pune apoi discul pe masa, si foloseste un cutter ascutit si ingust pentru a taia un buzunar in interiorul discului.


disc din burete
buzunar realizat cu un cutter ingust

Realizeaza sase astfel de discuri, cate unul pentru fiecare senzor. In buzunarul taiat cu cutter, baga un senzor piezo. Testeaza in interfata de debug ca atunci cand bati in discul din burete sa vezi valori care se modifica corespunzator. Dupa ce te-ai convins ca senzorul functioneaza corect, poti sigila buzunarul cu un lipici pentru plastic sau cu banda dublu adeziva.

Vei obtine in final sase astfel de discuri de burete, fiecare disc din burete avand inglobat un senzor piezo in interior. Mai verifica inca o data ca atunci cand lovesti fiecare disc in parte, vezi informatiile corespunzatoare in interfata de debug. Foloseste programul de mai jos pentru acest test.

```
#define PRAG 3
void setup() {
      Serial.begin(9600);
}
void loop() {
      for (int i = 0; i < 5; i++) {
            int v = analogRead(i);
            if (v > PRAG) {
                  Serial.print("DISC :");
                  Serial.print(i);
                  Serial.print(";");
                  Serial.print("VAL :");
                  Serial.println(v);
            }
      }
}
```

Fata de varianta anterioara, in care afisam doar senzorul conectat la portul analogic zero in interfata de debug, acum verificam toti cei sase senzori pe rand, si daca vreunul dintre ei depaseste valoarea de prag, atunci il afisam in interfata de debug.

Daca totul este OK pana in acest punct, este momentul sa montezi peste Arduino si Music Instrument Shield. Codul sursa complet este mai jos.

```
#include <SoftwareSerial.h>

#define NOTE_DURATION 800
#define DELAY_PER_SOUND 100
#define SENSOR_COUNT 6
#define LED_PIN  13
#define MIDI_RESET_PIN 4


byte string[8] = {35, 38, 49, 45, 75, 76, 77, 85};

boolean inProgress[SENSOR_COUNT];
long sum[SENSOR_COUNT];
long count[SENSOR_COUNT];
int values[SENSOR_COUNT];
long timeStart[SENSOR_COUNT];

long playStartTime[SENSOR_COUNT];

SoftwareSerial mySerial(2, 3);


void setup() {
```

```
  Serial.begin(9600);
  Serial.println("BEGIN");
  for (int i=0; i<SENSOR_COUNT; i++){
    inProgress[i] = false;
    sum[i] = 0;
    count[i] = 0;
    playStartTime[i] = 0;
  }

  mySerial.begin(31250);


  pinMode(MIDI_RESET_PIN, OUTPUT);
  digitalWrite(MIDI_RESET_PIN, LOW);
  delay(100);
  digitalWrite(MIDI_RESET_PIN, HIGH);
  delay(100);

  talkMIDI(0xB0, 0, 0x78);

}

void loop() {
  checkStop();

  for (int i = 0; i < SENSOR_COUNT; i++){
    values[i] = analogRead(i);

    if (values[i] > 10) {
      if (!inProgress[i]){
        inProgress[i] = true;
        sum[i] = 0;
        count[i] = 0;
        timeStart[i] = millis();
      }
    }

    if (inProgress[i]) {
      sum[i] = sum[i] + values[i];
      count[i] = count[i] + 1;
      if ((values[i] <= 10)&&((millis() - timeStart[i]) > DELAY_PER_SOUND))
{
        inProgress[i] = false;
        doPlay(i, sum[i]);
      }
    }
  }
}



void checkStop() {
  for (int i = 0; i < SENSOR_COUNT; i++){
    if ((millis() - playStartTime[i]) > NOTE_DURATION) {
      playStartTime[i] = 0;
      noteOff(0, string[i] ,127);
    }
  }
```

```
}

void doPlay(int index, double strength) {
  Serial.print(index);
  Serial.print("-");
  Serial.println(strength);

  noteOn(0, string[index], 127);
  playStartTime[index] = millis();

}


void noteOn(byte channel, byte note, byte attack_velocity) {
  talkMIDI( (0x90 | channel), note, attack_velocity);
}


void noteOff(byte channel, byte note, byte release_velocity) {
  talkMIDI( (0x80 | channel), note, release_velocity);
}


void talkMIDI(byte cmd, byte data1, byte data2) {
  digitalWrite(LED_PIN, HIGH);
  mySerial.write(cmd);
  mySerial.write(data1);


  if( (cmd & 0xF0) <= 0xB0) {
    mySerial.write(data2);
  }
  digitalWrite(LED_PIN, LOW);

}
```

Codul de mai sus se bazeaza pe exemplul dat in sectiunea despre Musical Instrument Shield, asa ca daca vrei sa-ti amintesti cum anumite se foloseste Musical Instrument Shield, iti recomand sa recitesti aceasta sectiune.

Constanta NOTE_DURATION defineste cat de mult dureaza sunetul dupa ce unul dintre discuri a fost lovit. Am ales o durata de 800 de milisecunde, dar poti experimenta cu valoarea aceasta dupa cum iti place tie. Dupa ce a fost lovit un disc, vreme de NOTE_DURATION milisecunde se genereaza sunetul. Functia *checkStop*, apelata imediat la inceputul functiei *loop* verifica pe rand fiecare senzor daca nu cumva a trecut NOTE_DURATION milisecunde de cand a inceput sunetul asociat acelui senzor. Daca a trecut, atunci opreste sunetul. Imediat dupa ce a fost apelata functia *checkStop* in *loop*, urmeaza un ciclu *for* care verifica fiecare senzor pe rand. Daca valoarea citita pe unul dintre senzori depaseste o valoare prestabilita, inseamna ca cercul de burete respectiv a fost lovit si atunci se genereaza sunetul corespunzator.

Aceasta a fost lectia 6. In final, as vrea sa te rog sa ne oferi feedback asupra acestei lectii, pentru a ne permite sa le facem mai bune pe urmatoarele.

Este vorba despre un sondaj cu 4 intrebari (oricare este optionala), pe care il poti accesa dand click aici.

Sau ne poti contacta direct prin email la contact@robofun.ro .

Iti multumim,

Echipa Robofun.RO